AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the captioned patent application:

Listing of Claims:

1. (Previously Presented) A method of instrumenting a Common Object Model (COM) object invoked by a client for performing a selected business logic, comprising:

intercepting a request from the client for creating said COM object,

generating a wrapper object corresponding to said requested COM object, said wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object,

providing said client with a reference pointer to said wrapper COM object,

upon invocation of a method associated with an interface of the requested COM object by the client, invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said rinvoked method,

wherein said invoked virtual function references instructions for saving a start time marker, executing instructions corresponding to said invoked method, and saving a stop time marker upon completion of execution of said instructions associated with the invoked method.

2. (Previously Presented) The method of claim 1, further comprising registering said invoked method with an Application Response Measurement (ARM) agent upon invocation of said method by the client.

3. (Original) The method of claim 2, wherein saving a start time marker comprises invoking said ARM agent for generating a record for a transaction corresponding to invocation of said method of the requested COM object.

4. (Original) The method claim 3, wherein saving a stop time marker comprises invoking said ARM agent to generate said stop time marker.

5. (Original) The method of claim 1, wherein said wrapper COM object comprises a data structure for storing a number of arguments and a type of each argument associated with each method of said requested COM object.

6. (Canceled)

7. (Original) The method of claim 1, wherein said wrapper object comprises a reference pointer for referring to said universal interface.

8. (Previously Presented) The method of claim 1, further comprising

defining a policy that indicates whether to perform said step of generating a wrapper COM object corresponding to said requested COM object; and

determining whether to perform said step of generating a wrapper COM object using said policy.

9. (Previously Presented) The method of claim 8, wherein said policy identifies a requested proxy object, a requested COM object belonging to an MTS package, and a requested COM^{+} object for wrapping.

10. (Original) The method of claim 8, further comprising

storing said policy in a tabular format in a registry of a system on which said COM objects are executed.

11. (Original) The method of claim 1, wherein the step of intercepting a request comprises

patching code associated with one or more selected system functions.

12. (Original) The method of claim 11, wherein said system functions are provided in a dynamic link library.

13-14. (Canceled)

15. (Original) The method of claim 11, further comprising

for each of said selected system functions, utilizing a hook associated therewith to refer to a program for patching said system function.

16. (Original) The method of claim 15, wherein said hook comprises a designated string stored in system registry.

17. (Original) The method of claim 16, wherein said patching of the system function comprises

replacing selected bytes in a code corresponding to said system function with a jump instruction to a code for creating said requested COM object and generating said wrapper object.

18. (Original) The method of claim 17, further comprising

copying instructions in said system function code corrupted by said inserted jump instruction to an allocated data area.

19. (Original) The method of claim 18, further comprising

decoding said selected bytes prior to their replacement by said jump instruction to determine a total number of bytes corresponding to instructions corrupted by said inserted jump instruction.

20. (Currently Amended) A computer system having memory storing instructions for monitoring instrumenting a response time of a transaction performed by one or more Common Object Model (COM) objects invoked by a client and executing on one or more platforms, comprising:

one or more monitoring agents deployed on selected ones of said platforms, each monitoring agent configured to intercept intercepting a request for creating at least one of said COM objects generating and to generate a wrapper object corresponding to said requested COM object, wherein said wrapper object implements a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object. ~~object~~, said monitoring agent further configured to provide the client with a pointer to said wrapper object,

wherein upon invocation of a method associated with an interface of the requested COM object, a virtual function of said universal interface of said wrapper object, indexed by a number corresponding to an index number of said invoked method, is invoked,

wherein said invoked virtual function references instructions for instrumenting said invoked method.

21. (Previously Presented) The system of claim 20, further comprising

an Application Response Measurement (ARM) agent in communication with said monitoring agent.

22. (Original) The system of claim 21, wherein each of said virtual functions refers to a code having instructions for invoking said ARM agent prior to execution of an invoked method of a wrapped COM object to save a start time marker, executing said invoked method, and invoking said ARM agent subsequent to completion of execution of said invoked method to save a stop time marker.

23. (Currently Amended) A computer readable medium storing instructions for performing a method of instrumenting a Common Object Model (COM) object invoked by a ~~client comprising~~ client, comprising:

intercepting a request from the client for creating said COM object,

generating a wrapper object corresponding to said requested COM object, said wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object,

providing said client with a reference pointer to said wrapper COM object~~. object~~, and

upon invocation of a method associated with an interface of the requested COM object, invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said invoked method,

wherein said invoked virtual function references instructions for instrumenting said invoked method.